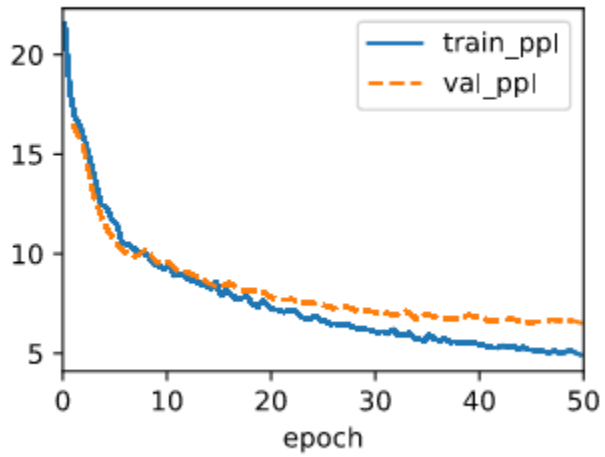
ECGR-5103: Homework 4
Patrick Flynn | ID: 801055057
29 March 2023
Github: https://github.com/pflynn157/ecgr-5106

Part 1

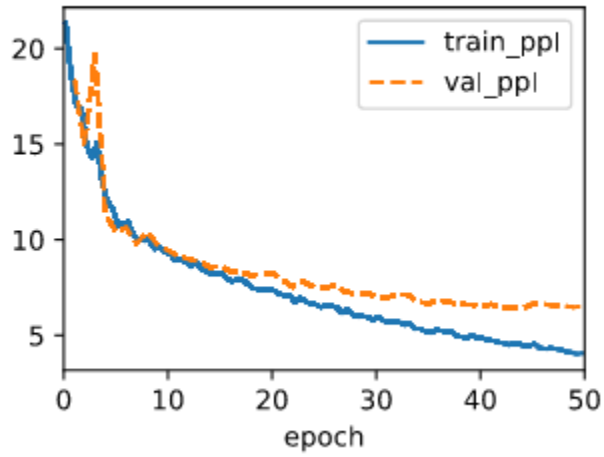
Problem A: The base model was the GRU model with the number of inputs being equal to the length of the vocabulary and 32 hidden layers. The baseline model did okay. The training and validation perplexity started out fairly high but dropped rapidly. Specifically, the training perplexity stayed lower than the validation perplexity. Running an example prediction returned a phrase that was grammatically correct but with words that didn't make sense.

```
'it has a conter and the ti'
```



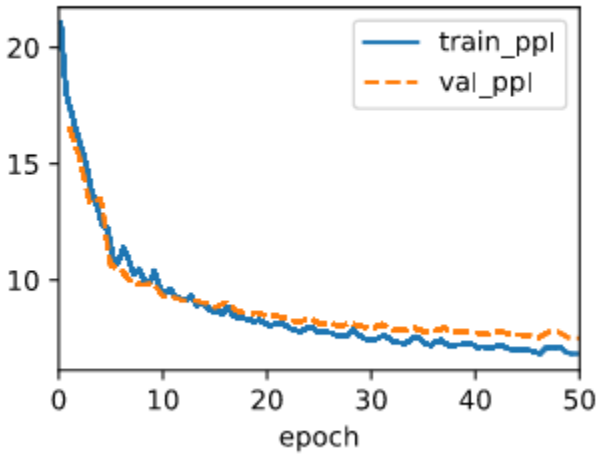
We tried two adjustments on the model, the first being an increase in the hidden layers up to 64 and the second a decrease down to 12. The results of this were interesting. In the latter, the training and validation perplexity was much lower, but the former returned a better prediction.
With 64 layers:

```
'it has in the time travell'
```



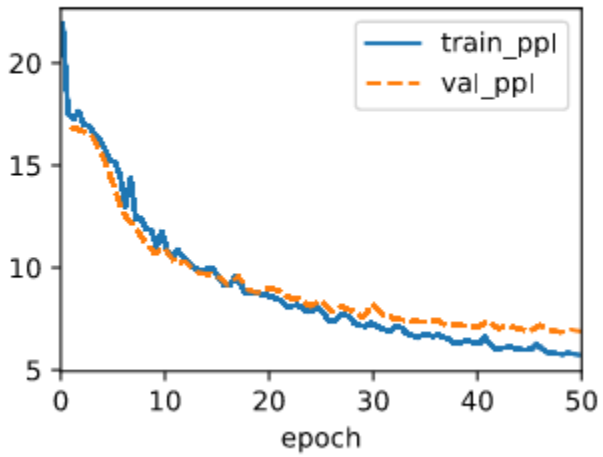
With 12 layers:

```
'it has the the the the the'
```



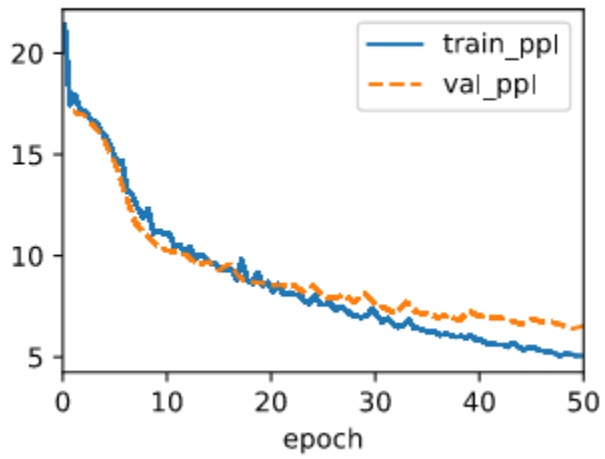
Problem B: In this part, we used the LTSM model. The default model with 32 hidden layers achieved a low level of training and validation perplexity, but yielded a poor prediction. These are the results of the baseline model:

```
'it has and the the that th'
```



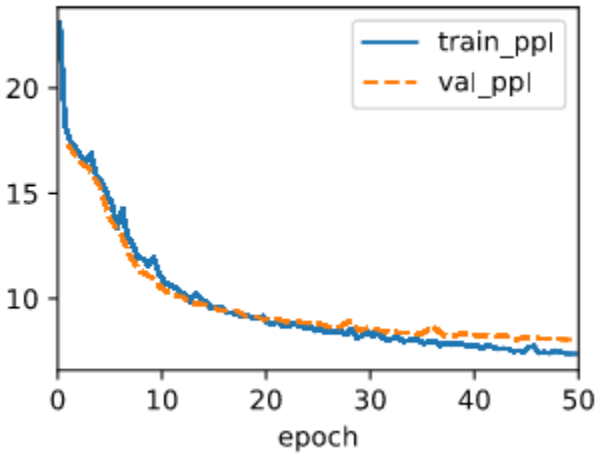
Once again, we adjusted the model to contain 64 hidden layers in the first test, and 12 hidden layers in the second test. The former once again resulted in slightly higher training and validation complexity, but it returned the best prediction yet.
With 64 layers:

```
'it has the time traveller '
```



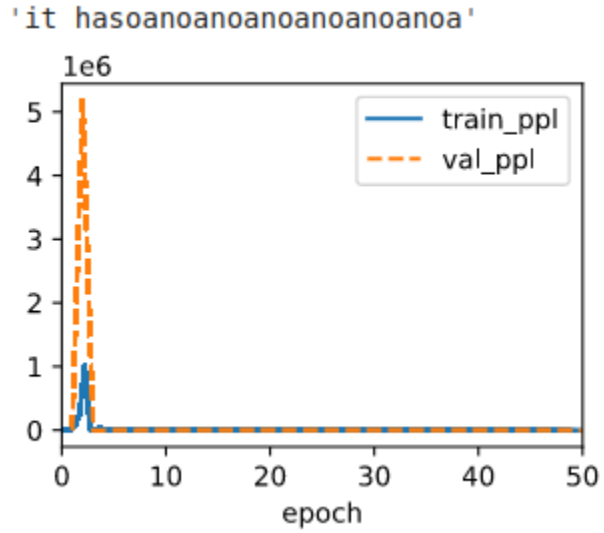
With 12 layers:

```
'it has the the the the the'
```

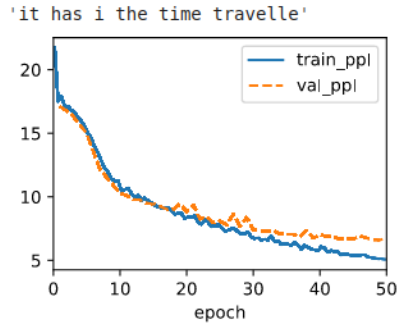


Problem C: In this section, we compared our best LSTM and GRU models with an RNN model, and checked the perplexity and predictions. Even though the perplexity was higher, 64 hidden layers seemed to yield a better prediction, so we did that. The LSTM and GRU models achieved similar results once again, but the RNN did very poorly. The perplexity was low, but the prediction was terrible.

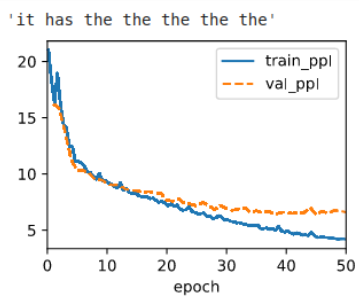
The RNN:



The LSTM:

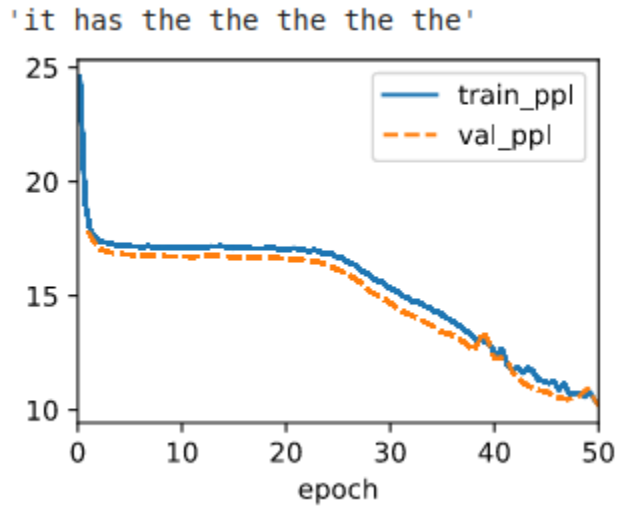


The GRU:



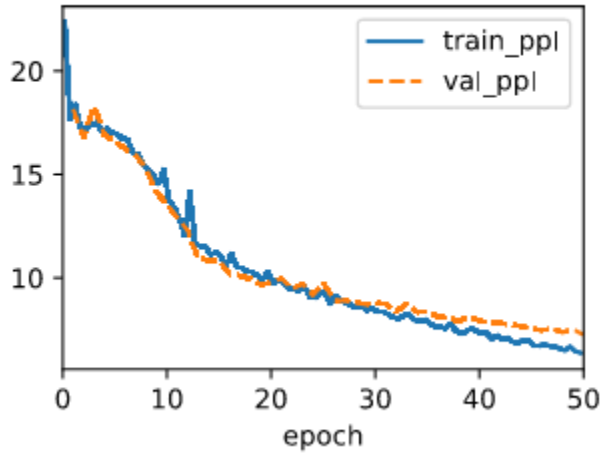
Part 2:

Problem A: In this section, we applied LSTM and GRU models to a deep neural network to see how this impacted the training and use of our data. In this problem, we first tested a baseline LSTM model with 32 hidden layers 2 layers. The training and validation perplexity was higher than before, and only decreased near the later epochs. The prediction was worse than former LSTM models.



Next, we created a baseline GRU model with 32 hidden layers and 2 layers. This performed much better than the LSTM model. The training and validation perplexities were much lower, and the prediction, while still not great, was better and made more sense.

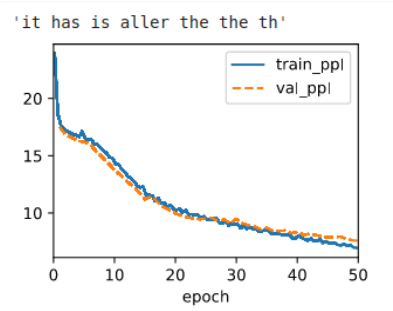
```
'it has the the time time t'
```



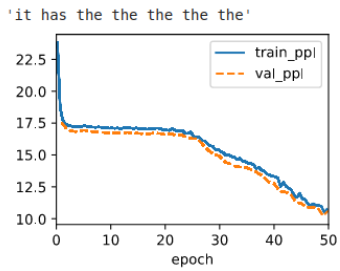
Next, we made several adjustments to the LSTM layer. We performed a total of six adjustments, divided into two parts. In the first part, we assigned 64 hidden layers with 1, 2, and 4 layers. In the second part, we assigned 12 hidden layers, with 1, 2, and 4 layers respectively.

64 hidden layers with 1 layer performed the best in terms of perplexity. As far as predictions go, the 64 w/ 1 model gave the most diverse prediction, even though it didn't make a ton of sense.

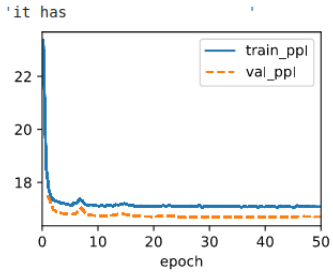
64 w/1



64 w/2

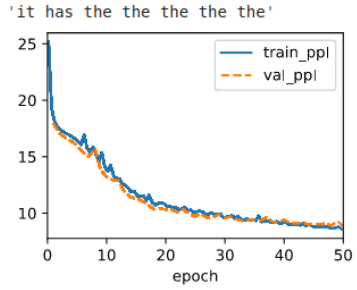


64 w/4

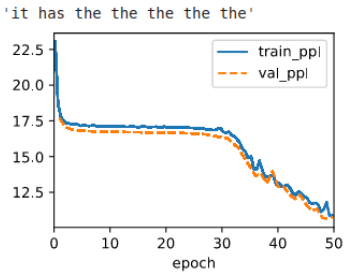


The results with 12 hidden layers were similar to that with 64, even with the number of layers adjusted. The perplexity with 1 layer was the lowest, but none of the predictions were very good.

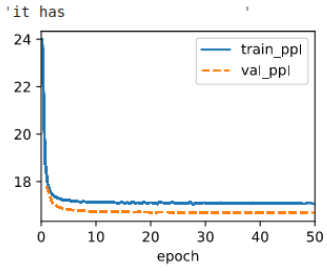
12 w/



12 w/2



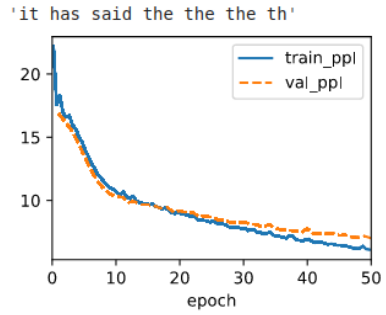
12 w/4



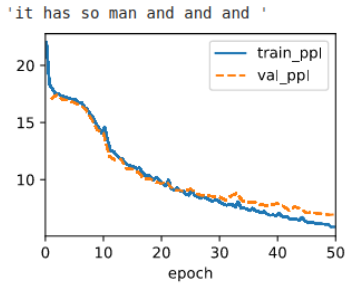
We made a similar adjustment to the GRU layer- 6 adjustments, divided into 2 parts, with part 1 containing 64 hidden layers and part 2 containing 12 hidden layers, and then three tests within each containing 1, 2, and 4 layers.

With 64 hidden layers, the test with 1 layer did the best in terms of perplexity and predictions. The test with 2 layers didn't do too poorly, but the one with 4 layers did poorly across the board.

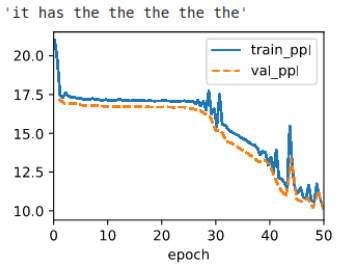
64 w/1



64 w/2

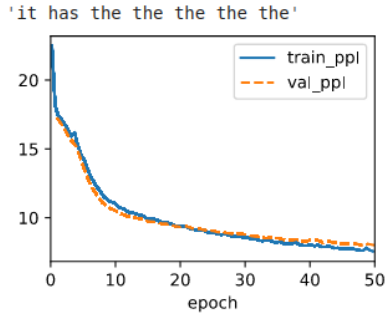


64 w/4

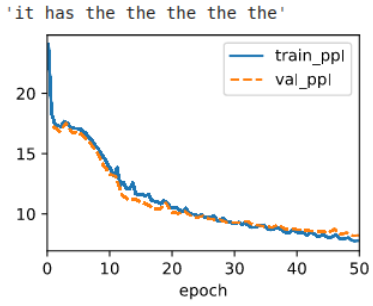


With 12 hidden layers, none of the results were that great. Those with 1 and 2 layers did okay in terms of perplexity, but the predictions weren't that great. The one with 4 layers was just in general poor.

12 w/1



12 w/2



12 w/4